NEW ES6 HOTNESS

ES6 = ECMASCRIPT 6

ECMAScript 6 is the 6th edition of ECMA-262, the standard defining the behaviors of the JavaScript language.

JavaScript is an implementation of ECMAScript, as are JScript and ActionScript.

“ECMASCRIPT WAS ALWAYS AN UNWANTED TRADE NAME THAT SOUNDS LIKE A SKIN DISEASE.”

Brendan Eich

ECMA = EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION[1]

They are a standards organization, like ISO.

[1] (but that doesn't matter)

ECMASCRIPT EDITIONS

» 1 - June 1997

» 2 - June 1998

» 3 - December 1999

» 4 - Oops...

» 5 - December 2009

» 6 - Mid-2015

» 7 - TBA

EXCITING NEW FEATURES

FAT ARROWS

```
var cat = name => console.log("my cat is named " + name);
cat('spot');
```

same as

```
var cat = function (name) {
  return console.log("my cat is named " + name);
};
cat("spot");
```

MORE ON FAT ARROWS

```
var numbers = [ 5, 4, 3, 2, 1];

var song = numbers.map( n => n + " bottles of beer on the wall" );

console.log( song );

/*
[ '5 bottles of beer on the wall',
  '4 bottles of beer on the wall',
  '3 bottles of beer on the wall',
  '2 bottles of beer on the wall',
  '1 bottles of beer on the wall' ]
*/
```

CLASSES

```
class FarmAnimal {
  constructor() {
    this.legs = 4
  }
}

class Cow extends FarmAnimal {
  speak() {
    console.log("moo. I have " + this.legs + " legs.")
  }
}

var c = new Cow();
c.speak(); // moo. I have 4 legs.
```

FANCY STRING INTERPOLATION

```
var legs = 7; // mutant cow
console.log(`I have ${legs} legs.`); // I have 7 legs
```

Note the back-ticks (`) which are not the same as (') or (").

ENHANCED OBJECTS {}

```
var a = 'apple';
var b = 'banana';

var fruit = { a, b, c: 'cherry' };

console.log( fruit ); // { a: 'apple', b: 'banana', c: 'cherry' }
```

DEFAULT PARAMS

```
function speak(phrase="Um") {
  console.log(phrase);
}
```

is nicer than the es5 equivalent

```
function speak(phrase) {
  phrase = phrase || "Um";
  console.log(phrase);
}

speak(); // Um
```

DESTRUCTURING

```
var a = "a";
var b = "b";

[a,b] = [b,a];

console.log(a); // b
```

SPREAD OPERATOR + FOR..OF LOOPS

```
function makeWords(word, ...suffixes) {
  for (var suffix of suffixes) {
    console.log( word + suffix );
  }
}

makeWords('bake', 'ry', 'r', 'd');
/*
bakery
baker
baked
*/
```

LET AND VARIABLE SCOPE

flagpole scope

```
{
  var cat = 'meow';
}
console.log(cat); // meow
```

block scope

```
{
  let dog = 'woof';
}
console.log(dog); // ReferenceError: dog is not defined
```

MODULES

lib/egyptianMath.js

```
export var pi = 22/7;
export function cubitsToFeet(cb) {
  return cb * 1.5;
}
```

someFile.js

```
import {pi} from 'lib/egyptianMath'
console.log(pi);
```

otherFile.js

```
import * as eMath from 'lib/egyptianMath'
console.log(eMath.pi);
```

SETS

```
var s = new Set();

s.add("OMG")
s.add("!")
s.add("ZOMG")
s.add("!")
s.add("!")
s.add("!")

console.log(s.size); // 3
```

PROXIES [2]

```
var pets = [ 'cat', 'dog', 'fish', 'elephant' ];

console.log( pets[-1] ); // undefined

var superPets = new Proxy(pets, {
  get: function(object, index) {
    if (Number(index) < 0) {
      return object[ object.length + Number(index) ];
    } else {
      return object[ index ];
    }
  }
});

console.log( superPets[-1] ); // "elephant"
```

[2] Not supported by transpilers

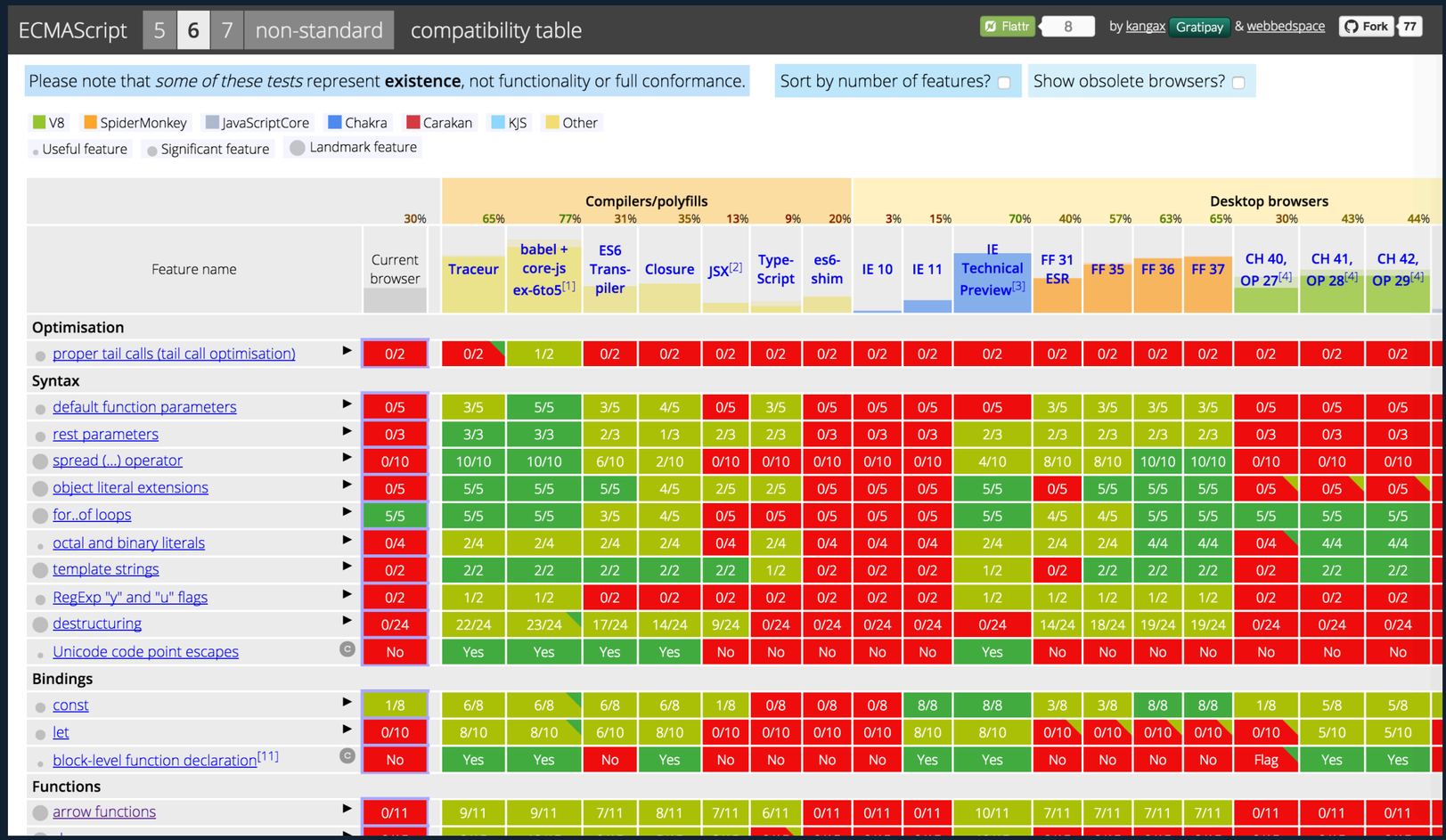
GENERATORS

```
function *fibonacci() {
        var a=1, b=1;
        while (true) {
                [a,b] = [b,a+b];
                yield b;
        }
}

var fib = fibonacci();

console.log( fib.next() ); // { value: 2, done: false }
console.log( fib.next() ); // { value: 3, done: false }
console.log( fib.next() ); // { value: 5, done: false }
console.log( fib.next() ); // { value: 8, done: false }
console.log( fib.next() ); // { value: 13, done: false }
```

PROMISES

```
var promise = new Promise(function(resolve,reject) {
  setTimeout(function() {
    if ( Math.random() > 0.5 ) {
      resolve('alive');
    } else {
      reject('dead');
    }
  }, 500);
});

promise.then(
  function(value) {
    console.log('Yay, the cat is ' + value);
  },
  function(error) {
    console.log('Sorry, the cat is ' + error);
  }
);
```

HOW DO I USE THIS STUFF?



ECMAScript 5 6 7 non-standard compatibility table

Flattr 8 by kangax Gratipay & webbedspace Fork 77

Please note that *some of these tests* represent **existence**, not functionality or full conformance.

Sort by number of features? Show obsolete browsers?

V8 SpiderMonkey JavaScriptCore Chakra Carakan KJS Other

Useful feature Significant feature Landmark feature

		Compilers/polyfills							Desktop browsers									
	30%	65%	77%	31%	35%	13%	9%	20%	3%	15%	70%	40%	57%	63%	65%	30%	43%	44%
Feature name	Current browser	Traceur	babel + core-js ex-6to5[1]	ES6 Trans-piler	Closure	JSX[2]	Type-Script	es6-shim	IE 10	IE 11	IE Technical Preview[3]	FF 31 ESR	FF 35	FF 36	FF 37	CH 40, OP 27[4]	CH 41, OP 28[4]	CH 42, OP 29[4]
Optimisation																		
proper tail calls (tail call optimisation)	0/2	0/2	1/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2
Syntax																		
default function parameters	0/5	3/5	5/5	3/5	4/5	0/5	3/5	0/5	0/5	0/5	0/5	3/5	3/5	3/5	3/5	0/5	0/5	0/5
rest parameters	0/3	3/3	3/3	2/3	1/3	2/3	2/3	0/3	0/3	0/3	2/3	2/3	2/3	2/3	2/3	0/3	0/3	0/3
spread (...) operator	0/10	10/10	10/10	6/10	2/10	0/10	0/10	0/10	0/10	0/10	4/10	8/10	8/10	10/10	10/10	0/10	0/10	0/10
object literal extensions	0/5	5/5	5/5	5/5	4/5	2/5	2/5	0/5	0/5	0/5	5/5	0/5	5/5	5/5	5/5	0/5	0/5	0/5
for..of loops	5/5	5/5	5/5	3/5	4/5	0/5	0/5	0/5	0/5	0/5	5/5	4/5	4/5	5/5	5/5	5/5	5/5	5/5
octal and binary literals	0/4	2/4	2/4	2/4	2/4	0/4	2/4	0/4	0/4	0/4	2/4	2/4	2/4	4/4	4/4	0/4	4/4	4/4
template strings	0/2	2/2	2/2	2/2	2/2	2/2	1/2	0/2	0/2	0/2	1/2	0/2	2/2	2/2	2/2	0/2	2/2	2/2
RegExp "y" and "u" flags	0/2	1/2	1/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	1/2	1/2	1/2	1/2	1/2	0/2	0/2	0/2
destructuring	0/24	22/24	23/24	17/24	14/24	9/24	0/24	0/24	0/24	0/24	0/24	14/24	18/24	19/24	19/24	0/24	0/24	0/24
Unicode code point escapes	No	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	No	No	No	No	No	No	No
Bindings																		
const	1/8	6/8	6/8	6/8	6/8	1/8	0/8	0/8	0/8	8/8	8/8	3/8	3/8	8/8	8/8	1/8	5/8	5/8
let	0/10	8/10	8/10	6/10	8/10	0/10	0/10	0/10	0/10	8/10	8/10	0/10	0/10	0/10	0/10	0/10	5/10	5/10
block-level function declaration[11]	No	Yes	Yes	No	Yes	No	No	No	No	Yes	Yes	No	No	No	No	Flag	Yes	Yes
Functions																		
arrow functions	0/11	9/11	9/11	7/11	8/11	7/11	6/11	0/11	0/11	0/11	10/11	7/11	7/11	7/11	7/11	0/11	0/11	0/11

HTTP://KANGAX.GITHUB.IO/COMPAT-TABLE/ES6/

IE TECHNICAL PREVIEW
70% SUPPORTED[3]

[3] Have to enable experimental web features.

FIREFOX 37
65% SUPPORTED

10.JS

42% SUPPORTED[4]

[4] With --es_staging flag

BABEL

77% SUPPORT

BABEL

Compiles ES6 code into ES3 code that your JS engine can run today.

```
$ cat interpolation.js

var legs = 7;
console.log(`I have ${legs} legs.`);

$ babel interpolation.js

"use strict";

var legs = 7;
console.log("I have " + legs + " legs.");
```

USING BABEL

```
npm install --g babel
```

enables

```
babel someFile.js # Outputs the generated ES3 code

babel-node someFile.js # Just like the node REPL
```

GRUNT

```
require("load-grunt-tasks")(grunt); // npm install --save-dev load-grunt-tasks

grunt.initConfig({
  "babel": {
    options: {
      sourceMap: true
    },
    dist: {
      files: {
        "dist/app.js": "src/app.js"
      }
    }
  }
});
```

BABEL AS A LIBRARY

main.js

```
require('babel/register');
var myApp = require('app');

// No ES6 code allowed in this file yet
myApp.run();
```

app.js

```
// ES6 code OK in this file and any else that it includes
module.exports = {
    run: function() {
      var numbers = [ 5, 4, 3, 2, 1];
      var song = numbers.map( n => n + " bottles of beer on the wall" );
      console.log( song );
    }
}
```

IN THE BROWSER

Probably not a great idea, but you could you know...

```
<script src="node_modules/babel/browser.js"></script>
<script type="text/babel">
class Test {
  test() {
    return "test";
  }
}

var test = new Test;
test.test(); // "test"
</script>
```

...AND A BUNCH OF OTHER PLACES.

» rails

» broccoli

» browserify

» brunch

» duo

» gobble

» gulp

RESOURCES

ECMAScript 6 compatibility table
http://kangax.github.io/compat-table/es6/

ECMAScript 2015 Language Specification
https://people.mozilla.org/~jorendorff/es6-draft.html

Overview of ECMAScript 6 features
https://github.com/lukehoban/es6features

QUESTIONS?